



# **Entrega final de obligatorio**

Federico Becoña, Rodrigo López y Juan Manuel Valiño  
Junio 2020

Universidad Católica del Uruguay  
Facultad de ingeniería y tecnologías  
Sistemas operativos

# Tabla de Contenidos

<b>Información teórica</b>	<b>3</b>
<b>Análisis del problema</b>	<b>6</b>
Problemas similares de sistemas operativos	6
Recursos	7
Procesos	8
Descripción de criterios de optimización	9
Ordenamiento de criterios de optimización	9
<b>Alternativas de solución</b>	<b>10</b>
Mapeo de vehículos	10
Distribución de vehículos	11
Semáforos en vías	11
Justificación	16
Posibles implementaciones de la simulación	17
Cabinas como hilos	17
Vehículos como hilos	17
<b>Planificación del sistema</b>	<b>18</b>
Escenarios	18
Vehículos con prioridad de paso	18
Bloqueo de vías	19
<b>Diagramas de la solución</b>	<b>20</b>
<b>Diagramas de la implementación</b>	<b>24</b>
Diagrama de sincronización	
Diagrama UML	24
<b>Simulaciones</b>	<b>25</b>
Comprobación del comportamiento	27
Casos simulados	32
<b>Estructura de la implementación</b>	<b>37</b>
Clase: Cabina	37
Clase: Peaje	39
Clase: Planificador	40
Clase: Vehículo	40
<b>Conclusiones</b>	<b>42</b>
<b>Lista de referencias</b>	<b>43</b>

# Información teórica

A continuación detallamos la información teórica adquirida en el curso, la cual utilizamos para la comprensión y modelado de la solución del problema.

## Procesos

El tópico de procesos es uno de los temas que es de gran ayuda para explicar ciertas situaciones que pueden ocurrir en un peaje:

Un proceso no es más que una instancia de un programa en ejecución, incluyendo los valores actuales del contador de programa, los registros y las variables.

[...]

La idea clave es que un proceso es una actividad de cierto tipo: tiene un programa, una entrada, una salida y un estado. (Tanenbaum: 2009, p. 84)

Encontramos en los procesos un buen paralelismo con varias de las tareas que se realizan.

## Recursos

Los recursos pueden llegar tener varias interpretaciones en cómo los mismos son aplicados. Pueden ser tanto componentes de hardware, así como datos o información.

Para el caso particular de este trabajo, identificamos una serie de recursos de hardware que son de gran ayuda para conseguir el cometido de generar un flujo vehicular fluido, y que serán detallados más adelante.

## Sistema de multiprogramación

Se trata de conmutar la ejecución entre procesos en la CPU en un período reducido de tiempo.

Dado que dicha conmutación se lleva a cabo en milisegundos, esto genera que en un período de tiempo dado pueda trabajar en varios procesos diferentes, dando la sensación de paralelismo.

## Planificación apropiativa

Este tipo de planificación implica que el CPU ejecuta un proceso por un tiempo determinado, y al terminar dicho tiempo, saca el proceso que se encontraba en ejecución para tomar el siguiente.

## Planificación no apropiativa

A diferencia del caso apropiativo, en este caso el procesador permite que el proceso tome el tiempo que necesite hasta que, o bien se bloquee el proceso o el mismo termina.

## Algoritmos de planificación

Se utilizan con el fin de que múltiples procesos puedan compartir el uso de un CPU. Los mismos determinan, por ejemplo, si un proceso debe ceder su lugar a otro para que sea procesado.

Pueden dividirse en distintas categorías:

#### *Procesamiento por lotes*

- **FCFS** (*First-Come, First-Served*): en este caso, los procesos van siendo ejecutados por su orden de llegada a la cola de procesos listos. Su comportamiento es igual que el de una estructura FIFO.
- **SJF** (*Shortest Job First*): tiene en cuenta el tamaño de la última ráfaga de CPU que haya sido utilizada, y teniendo en cuenta eso selecciona entre los procesos en espera aquel que tenga una menor duración.
- **SRTN** (*Shortest Remaining Time Next*): intenta mejorar el SJF, diferenciándose en que para este caso se tiene en cuenta el proceso que tenga el menor tiempo restante.

#### *Interactivo*

- **Round-robin**: consta de la utilización de un *quantum*, que representa el tiempo máximo que puede pasar la CPU ejecutando un proceso. Si dado el caso, el proceso continúa ejecutándose al finalizar dicho tiempo, es quitado del procesador y enviado al final de la cola de procesos listos. Por el contrario, si el proceso tuviera una duración menor que el *quantum*, el proceso simplemente terminaría al bloquearse.
- **Event Driven**: para su implementación es necesario que cada proceso tenga una prioridad, seleccionando para procesar en la CPU el que esté listo con la prioridad más alta.
- **Múltiples colas de prioridad.**

### **Estados de un proceso**

Existen tres tipos de estados en los que puede llegar a estar un proceso:

- *En ejecución*: cuando se encuentra en este estado, el mismo está haciendo uso de la CPU.
- *Listo*: en este estado se encuentra pronto para ejecutarse. Se encuentra así pues tuvo que permitir que otro proceso se ejecute.
- *Bloqueado*: es el estado en el que se encuentran los procesos mientras no exista una interacción externa que les permita ejecutarse.

### **Hilos**

Como menciona Tanenbaum en su libro “Sistemas operativos modernos”, los hilos son una especie de “miniprosesos”.

Los hilos se diferencian principalmente de los procesos en que permiten tener varios funcionando en paralelo, que comparten el espacio de direcciones y los datos entre los mismos.

Teniendo en cuenta lo anterior, se puede decir que en un sistema multihilo, cualquier hilo del mismo puede leer, escribir o hasta borrar la pila de otros.

## Comunicación entre procesos

Para lograr ciertos cometidos, es sumamente necesario que los procesos tengan la posibilidad de comunicarse entre sí.

Y no solamente es necesaria para permitir el pasaje de información entre procesos diferentes, sino que también permite evitar problemas de sincronización entre los mismos.

## Sección crítica

La sección crítica de un proceso es aquella sección del código fuente del mismo en la cual se comparte información común. La particularidad de esto se encuentra en que, en el momento que un proceso está operando sobre esa misma sección, ningún otro puede hacer lo mismo.

## Semáforos

Fueron introducidos por Dijkstra en 1965. Constan de una variable entera y dos operaciones estándar: wait y signal.

- **Wait:** comprueba que el valor del semáforo es mayor que 0. En caso que así sea decrementa el valor y continúa. En caso contrario, cuando el valor es 0, el proceso se pone a dormir sin finalizar la operación de forma momentánea.
- **Signal:** incrementa el valor del semáforo indicado. Esto implica que en caso de haber algún proceso previo inactivo por un wait previo, el mismo se despertará y podrá completar la operación wait que quedó pendiente.

## Mutexes

Es un caso particular de un semáforo, diferenciándose en que en los mutex no se hace uso de la finalidad de contar.

Tiene dos estados:

- *Abierto o desbloqueado.*
- *Cerrado o bloqueado.*

Dichos estados se representan con un 0 en el caso de encontrarse cerrado y cualquier otro valor indica que está abierto.

Para acceder a una sección crítica se hace uso del *mutex lock*, pudiendo ingresar a la región si el mutex se encuentra abierta.

En caso de que se encuentre cerrado, el hilo que hizo la llamada se bloquea hasta que el hilo en la región crítica finalice y haga uso del *mutex unlock*.

# Análisis del problema

Mediante el estudio del problema y basándonos en la utilización de tecnología de comunicación remota, debemos proponer distintas estrategias, diseñar una solución y realizar una simulación para el ordenamiento del tránsito de un peaje de forma tal que se optimicen fundamentalmente los tiempos de paso de los vehículos. Además se pide que se tengan en cuenta otros factores que pueden ocurrir como, tránsito cortado, unidad de emergencia en situación crítica, preferencia de paso, etc, considerando el costo que genera darle importancia a estas distintas situaciones.

Contexto: La propuesta se enfoca en el funcionamiento del peaje Pando.

## Problemas similares de sistemas operativos

Un buen paralelismo entre el trabajo que estamos realizando y los problemas que ocurren en los sistemas operativos, es la forma en la que se manejan los procesos. Estos resuelven la ejecución de los procesos mediante el uso de distintos algoritmos de planificación. Analizando el problema del peaje a partir de estos algoritmos podemos identificar algunos comportamientos similares que son de utilidad para llegar al diseño de una solución, como por ejemplo:

- FCFS (First come, First Served): En este algoritmo los procesos se ejecutan en el orden que llegan a la cola de procesos listos. En el caso del peaje, esto se podría pensar como hacer pasar a los vehículos según su orden de llegada.
- Event Driven: Este algoritmo asigna prioridades a los procesos y estos se ejecutan teniendo en cuenta el orden de los mismos, tomando antes que nada el que tenga la prioridad más altas. Desde el punto del peaje, se podría hacer el paralelismo de que el pasaje de los vehículos por las cabinas tiene distinta prioridad dependiendo de si por ejemplo se trata de ambulancias, patrullas, bomberos, vehículos de carga o automóviles comunes. En base a esa jerarquía se los hace pasar antes o después.
- Colas de múltiples niveles: En este algoritmo se realiza una clasificación de los procesos, asignando los mismos a diferentes colas cada uno planificando en varios niveles, primero colas y luego proceso. En este caso se podría pensar el pasaje de vehículos por las cabinas según prioridades asignadas en base a distintas colas que a su vez le den mayor importancia a los procesos cuyos vehículos hayan llegado antes.

Para construir nuestra solución también podemos aprovechar la forma en la que los sistemas operativos resuelven la ejecución de los procesos haciendo uso de la concurrencia y la sincronización. Llevando estos conceptos al escenario del peaje se podría visualizar una competencia entre los procesos de pasaje de los vehículos. El procesamiento en las diferentes cabinas se da en forma simultánea de forma tal de que no puede pasar que dos cabinas procesen el cobro de un mismo automóvil en el mismo instante.

## Recursos

Logramos identificar los siguientes recursos en el peaje de Pando:

- **Cabinas de cobro:**  
Constan de un puesto de cobro por el cual pueden pasar de a un vehículo a la vez para realizar el pago de forma ordenada. Estas se encuentran equipadas con varios de los modernos recursos detallados posteriormente.
- **Etiquetas inteligentes (TAG):**  
Son pequeñas etiquetas con un número de serie único, que van pegadas en los parabrisas de los vehículos con el objetivo de que el sistema los reconozca como usuarios registrados.
- **Antena para el reconocimiento de los TAGs de los vehículos:**  
Estos dispositivos están instalados en cada una de las cabinas con el fin de reconocer las etiquetas de los vehículos que pasan.
- **Cámaras de reconocimiento de matrículas:**  
Se encuentran en las cabinas de cobro con el objetivo de registrar las matrículas de los vehículos que pasan por ella.
- **Barreras levadizas:**  
Mediante el uso de robótica las barreras se levantan o se bajan de acuerdo la necesidad del momento.
- **Infraestructura informática:**  
Todo el equipamiento informático necesario para el correcto funcionamiento del sistema.
- **Conexión a internet:**  
Se utiliza para procesar los pagos con los distintos entes de tarjetas de crédito o débito y notificar si un vehículo no pudo pagar ya sea por saldo insuficiente o porque hubo un problema con su tag para que se le realice el pago posteriormente.
- **Señalización vial:**  
Es necesario tener una señalización vial clara con el fin de evitar embotellamientos o accidentes.
- **Semáforos:**  
Utilizados con el objetivo de indicar a los vehículos si pueden hacer uso de una determinada cabina o si deben esperar.
- **Letreros led:**  
Su finalidad es colaborar en la distribución del tráfico indicando a los conductores cuál vía deben tomar.
- **Sensor de reconocimiento**  
Se utilizan para reconocer si un espacio está siendo ocupado por un vehículo
- **Sistema de cámaras con reconocimiento de vehículos**  
Cámaras situadas en varios puntos del peaje que interactúan con los sensores para generar calcular de qué forma están distribuidos los vehículos en el peaje.

## Procesos

- Cobro del monto correspondiente a cada usuario de vehículos:
  - Luego de identificar el vehículo que está pasando por la cabina, se utiliza esta información para verificar con la entidad correspondiente si el saldo del que dispone es suficiente para realizar el cobro o de lo contrario tomar medidas que eviten un enlentecimiento del tránsito.
- Reconocimiento de matrículas:
  - En caso de no tener saldo, que el tag no sea reconocido, o que no se cuente con un tag, el reconocimiento de la matrícula se utiliza para generar un cargo en alguna entidad como SUCIVE. Además, esta función es de utilidad para informar acerca de la ubicación de un vehículo con matrícula requerido por algún ente del estado.
- Levantar / bajar la barrera para permitir o no el pasaje de vehículos:
  - La barrera actuará dependiendo de si el vehículo está en condiciones de pasar o no. El intervalo de tiempo entre que la barrera se levanta y se baja debe ser suficiente para el vehículo habilitado pase sin que el inmediato anterior lo haga sin pagar.
- Direccionamiento del tránsito:
  - Con el objetivo de obtener un uso eficiente de las cabinas activas, se distribuyen los vehículos utilizando distintos recursos.



## Descripción de criterios de optimización

- **Maximizar la utilización de las cabinas en uso:** Se busca procesar la mayor cantidad de vehículos en el menor tiempo posible.
- **Dar prioridad a vehículos de emergencia:** Se atiende de la forma más rápida a un vehículo que cumple con determinadas condiciones, por ejemplo una emergencia.

## Ordenamiento de criterios de optimización

1. **Dar prioridad a vehículos de emergencia:** Consideramos que este es el criterio más importante ya que si bien el resto de los usuarios se ven perjudicados por la demora, el facilitar el pasaje de vehículo para colaborar con la situación de emergencia puede llegar a significar salvar vidas.
2. **Maximizar la utilización de las cabinas en uso:** Este es el criterio que más impacta en la optimización del tiempo utilizando de forma inteligente los recursos disponibles y evitando la sobrecarga del resto de las cabinas .

# Alternativas de solución

## Mapecto de vehículos

Para poder planificar la forma en la que se distribuirá el tráfico entre las distintas vías del peaje es necesario poder contar con la información de la carga actual de todas las vías en todo momento. Para solucionar esto es que proponemos una combinación de tecnologías de sensores (utilizados normalmente en los parkings inteligentes) así como el uso de cámaras con software de reconocimiento ubicadas en distintos puntos dentro y en las cercanías del peaje.



## Distribución de vehículos

En el punto en el que se amplía la ruta interbalnearia se utilizará 1 letrero led por cada carril el cual indicará a qué vía del peaje debe ir el vehículo.

Cada vía estará correctamente señalizada con su número en las cabinas asociadas a cada vía con letreros led que se pueden ver a gran distancia.

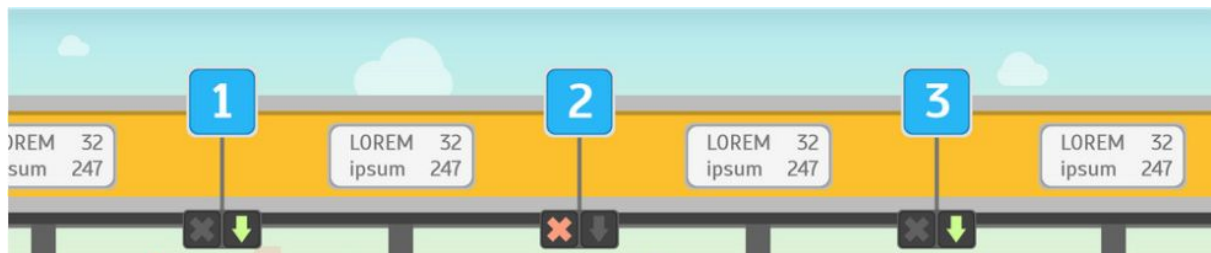
## Semáforos en vías

Cada vía tendrá su semáforo que indicará a los conductores si pueden pasar por la misma o no. Dicho semáforo se pondrá en rojo en caso de que alguna unidad de emergencia ingrese a la vía lo que indicará a los vehículos que no deben continuar. De esa forma las emergencias pasarán sin obstáculos y de forma más ágil.

Ejemplo de señalización en la entrada al Peaje:



Ejemplo de identificación de cabinas:



## Alternativa 1

En esta alternativa se utilizan todas las vías de igual forma para distribuir el tráfico.



- *Distribución del tráfico*

La distribución se realiza hacia las distintas vías que se abren desde la ruta y terminan en las cabinas en funcionamiento que no estén bloqueadas. Para ello se utiliza un sistema con una pantalla por carril de la ruta que indica la cabina a la que debe ir cada vehículo.

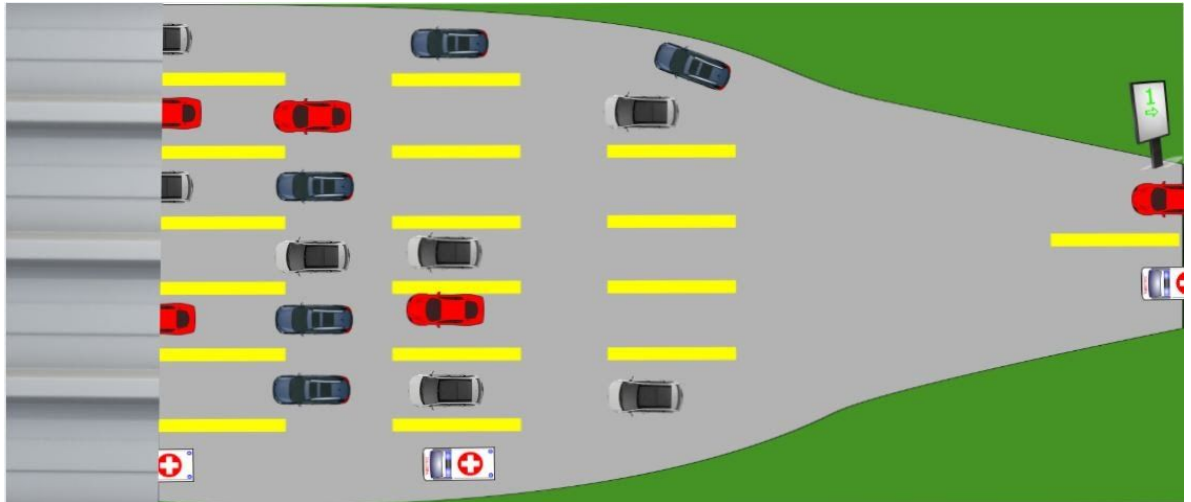
Cuando un vehículo llega a la zona del peaje es dirigido hacia la cola con menos vehículos logrando así una distribución uniforme en cuanto a la carga de todas las cabinas. Para poder medir los lugares disponibles en cada cola haremos uso de un mapeo de los automóviles que esperan en ellas.

- *Bloqueo en una vía*

Ante el bloqueo de un carril se deja de enviar nuevos vehículos a dicha esa vía. Posteriormente, se desvían los vehículos del carril bloqueado a las vías situadas en sus costados.

## Alternativa 2

En esta alternativa existen distintos tipos de vía, las normales que se utilizan para distribuir los vehículos sin prioridad, y la de emergencia que es usada para la atención exclusiva de los vehículos con prioridad.



- *Distribución del tráfico*

La distribución se realiza hacia las distintas vías que se abren desde la ruta y terminan en las cabinas en funcionamiento que no estén bloqueadas.

Para esta repartición se utiliza un sistema con una pantalla por cada carril de la ruta en la que se le comunica al conductor la vía a la que debe desplazarse.

- *Priorización del paso para los vehículos de emergencia*

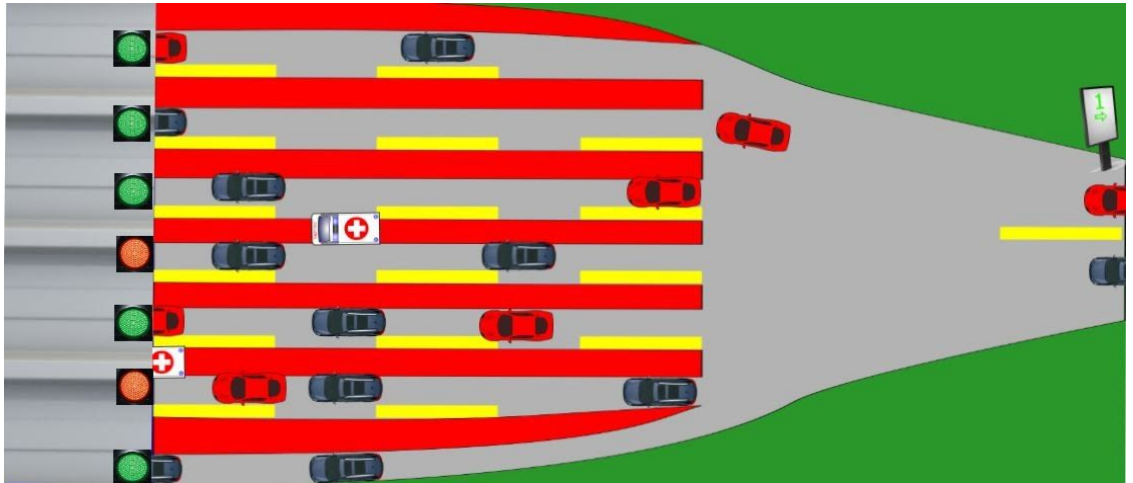
En la distribución del tráfico se diferencia entre vehículos de emergencia y normales. Los primeros son asignados a una vía exclusiva para ellos y los últimos se envían a una de las otras a partir del cálculo de cuál se encuentra menos cargada en dicho instante. De esta forma se intenta conseguir una distribución uniforme del tráfico entre las vías de las cabinas. El cómputo mencionado se hace usando un mapeo de los vehículos que hay en cada vía.

- *Bloqueo en una vía*

El sistema no permite la recepción de nuevos vehículos tanto al carril bloqueado como a las vías más próximas a ella. Luego se autoriza a todos los vehículos que estaban trancados a redistribuirse hacia ellas.

### Alternativa 3

En esta alternativa cada cabina tiene asociada una vía que cuenta con una subdivisión identificada por medio de un color indicando que por la misma solo pueden circular vehículos de alta prioridad.



- *Distribución del tráfico*

La distribución del tráfico se produce hacia los distintos carriles de las diferentes vías que se abren desde la ruta y terminan en las cabinas en funcionamiento sin estar bloqueadas.

Tampoco se asignan vehículos a un carril si, el carril del mismo tipo de una vía adyacente queda trancado, este es el que tiene menos automóviles (en el instante del bloqueo) de los carriles de ese tipo de los que se encuentran a los costados del carril trancado y él no está bloqueado. En este caso, se vuelven a asignar vehículos al carril mencionado tras haber repartido la totalidad de los automóviles trancados.

El reparto de vehículos se realiza en el punto en el que nacen las vías mencionadas usando un sistema con una pantalla por carril de la ruta.

En las pantallas se le comunica a cada conductor a qué carril debe dirigirse a partir del cálculo de cuál es la casilla menos cargada para su nivel de prioridad en ese momento. Así, se busca conseguir una distribución uniforme del tráfico entre los carriles de las vías. Este proceso se realiza utilizando un mapeo de los automóviles esperan en las distintas vías del peaje.

- *Priorización del paso para los vehículos de emergencia*

En cada una de las vías, próximo a su respectiva cabina, existe un semáforo que avisa a los vehículos que están por ingresar a la casilla si tienen o no permitido el paso. En el momento en que un vehículo de emergencia es asignado a una vía, el semáforo se enciende indicando a los conductores que esperan para acceder a dicha cabina que deben esperar a que se atienda primero al vehículo con prioridad de paso. Luego de que el vehículo es

atendido, si no hay más automóviles de prioridad en espera, el semáforo vuelve a cambiar y el tránsito por esta vía vuelve a la normalidad.

- *Bloqueo en una vía*

Se detiene la distribución de los vehículos que llegan al carril bloqueado de una vía ya sea normal o de emergencia. Además se deja de asignar tráfico a la vía adyacente del mismo tipo que la bloqueada, que tenga menos vehículos de esa clase y no esté bloqueada. De esta forma, se permite que la totalidad de los automóviles que estaban en la vía que ocurrió el bloqueo puedan redistribuirse a las vías contiguas.

Se toma en cuenta que un vehículo debe esperar para cambiarse de vía a tener un espacio para hacerlo en el nuevo carril y que también haya espacio en el carril que debe atravesar para llegar hasta allí de forma tal que no se genere un accidente.

## Justificación

Seleccionamos la alternativa tres ya que esta resulta óptima en cuanto a la atención de vehículos de emergencia (el criterio de optimización más importante para nosotros) y mantiene un equilibrio con el criterio de minimización los tiempos de paso del peaje.

La primera alternativa logra una óptima distribución de los vehículos en cuanto a minimización de los tiempos del peaje pero no cumple con el criterio de dar prioridad al paso de vehículos de emergencia.

La segunda mejora la atención de los vehículos de emergencia dedicando una vía exclusivamente para atenderlos a ellos. Sin embargo, esto trae como consecuencia que se pierda capacidad de procesamiento para los vehículos sin prioridad.

La tercera consigue la óptima capacidad de atención para vehículos de emergencia y se encuentra muy alineada con el criterio de maximizar la utilización de las cabinas.

Esta última alternativa es superior a la primera en cuanto a la atención de emergencias al atender dichos vehículos de inmediato. A cambio de ello recibe una leve penalización en la minimización de los tiempos del peaje debido a que se dejan cabinas sin utilizar por automóviles sin prioridad al llegar vehículos de emergencia.

Por otro lado, la opción elegida es más conveniente que la segunda en cuanto a la atención de emergencias al permitir el paso de más de un vehículo de prioridad simultáneamente y al no subutilizar ninguna vía perdiendo así capacidad de procesamiento en paralelo.



## Posibles implementaciones de la simulación

### Cabinas como hilos

Para la implementación de la simulación utilizaremos un planificador y tantos hilos como cabinas. Por otra parte, se medirán “unidades de tiempo” que son utilizadas para sincronizar las actividades de las cabinas y del planificador. Una “unidad de tiempo” representa que todos los hilos terminaron de ejecutar las instrucciones correspondientes a dicho “instante”.

Cada cabina tiene su propio arreglo de vehículos sin prioridad y su arreglo de vehículos de emergencia. Estos serán procesados a lo largo de los distintos instantes de tiempo.

Para representar la llegada de los vehículos al peaje se utiliza una cola que contiene todos los vehículos ordenados por sus respectivos tiempos de llegada.

### Vehículos como hilos

Se cuenta con un hilo planificador que se encarga de manejar las unidades de tiempo y múltiples hilos que representan los vehículos. En cada “unidad de tiempo”, por un lado, el planificador asigna cada vehículo que llega al peaje en ese momento a una cola de una cabina tomando en cuenta su nivel de prioridad. Por otro lado, el vehículo verifica si es su turno de pasar en la cola en la que fue asignado, si ocupa la cabina o si tiene que dejar pasar a otro automóvil, etc. Al ocupar la casilla, luego de que transcurre el tiempo correspondiente al paso por el peaje, la libera y la deja disponible para el uso de otros automóviles.

# Planificación del sistema

## Escenarios

### Vehículos con prioridad de paso

- Ante una emergencia, como podrían ser una ambulancia que traslada un paciente, un camión de bomberos o la policía caminera.
  1. En caso de reconocer un vehículo de emergencia el planificador establece la prioridad de paso del vehículo y lo asigna la vía disponible con la cola de prioridad que tenga menor tamaño.
  2. Si previamente no hay otros vehículos de emergencia en la cola a la que se lo asignó, inmediatamente se dispara un mecanismo que enciende el semáforo situado antes del acceso de la cabina, bloqueando así el paso de los vehículos normales que están esperando a ser procesados.
  3. El vehículo con emergencia avanza sin obstáculos hasta pasar por la cabina a la que fue asignado haciendo uso del espacio distinguido con color rojo por el cual no puede circular ningún otro vehículo que no sea de prioridad.
  4. Luego de que la emergencia es atendida el proceso concluye habilitando nuevamente el semáforo para que continúe la circulación normalmente.

Teniendo en cuenta que la penalización de tiempo no es muy alta durante el pasaje del automóvil con prioridad, el planificador puede eventualmente continuar asignándole vehículos a su vía dependiendo de la carga general de las demás.

## Bloqueo de vías

- Choques o desperfectos mecánicos de vehículos en alguna de los carriles de acceso a las cabinas.

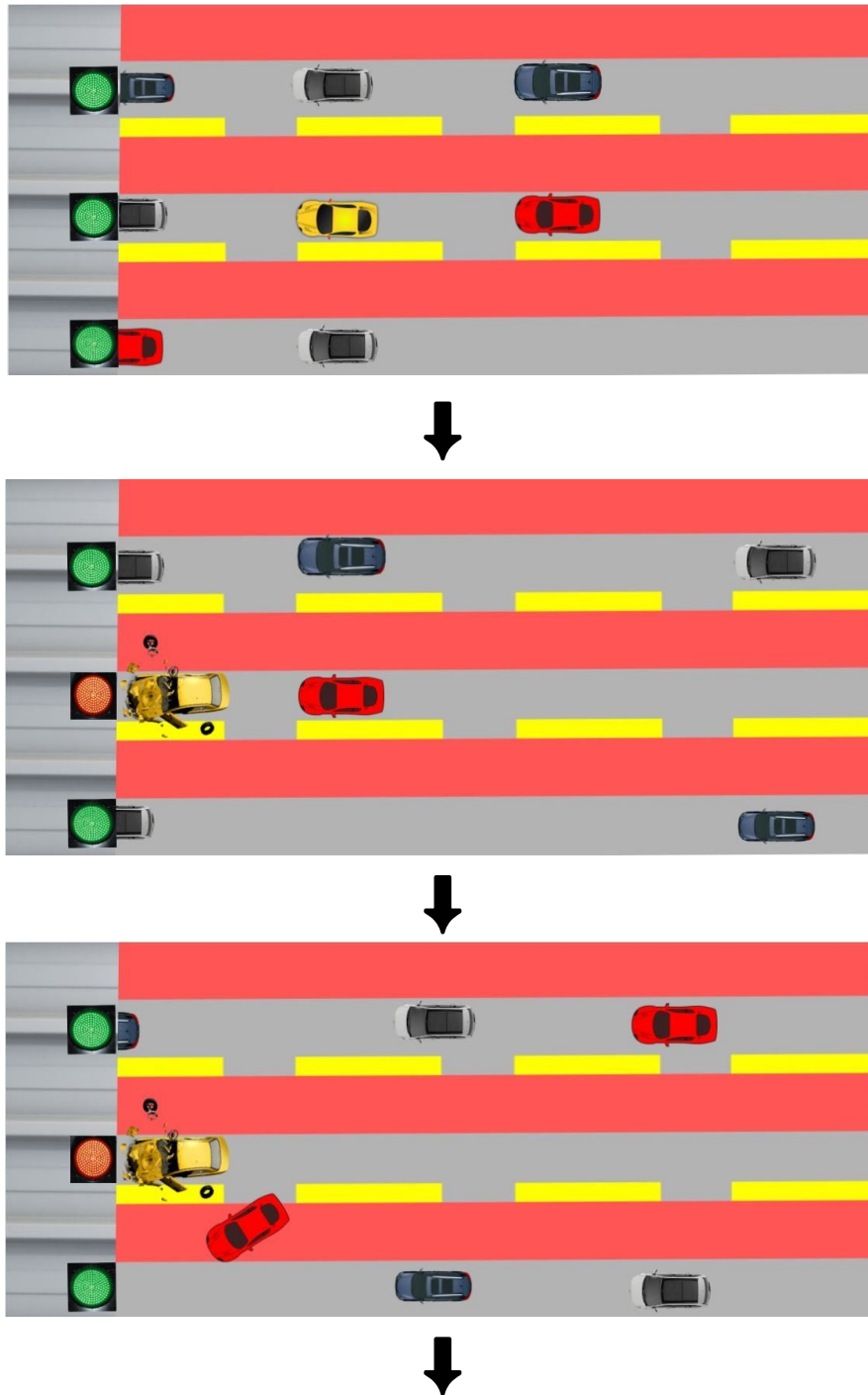
Ante el bloqueo de un carril el planificador actúa de forma rápida para redistribuir a los vehículos que ya se encontraban en ella. Para solucionar este problema tomamos en cuenta que los vehículos se encuentran imposibilitados de moverse hacia adelante siendo la única opción posible moverse hacia los costados, pues de otra forma tendrían que hacer una maniobra un tanto riesgosa que podría conducir a nuevos bloqueos. De esta forma, para que la simulación se asemeje lo más posible a la realidad, solo permitiremos que los vehículos que se encuentran trancados puedan moverse hacia las vías adyacentes.

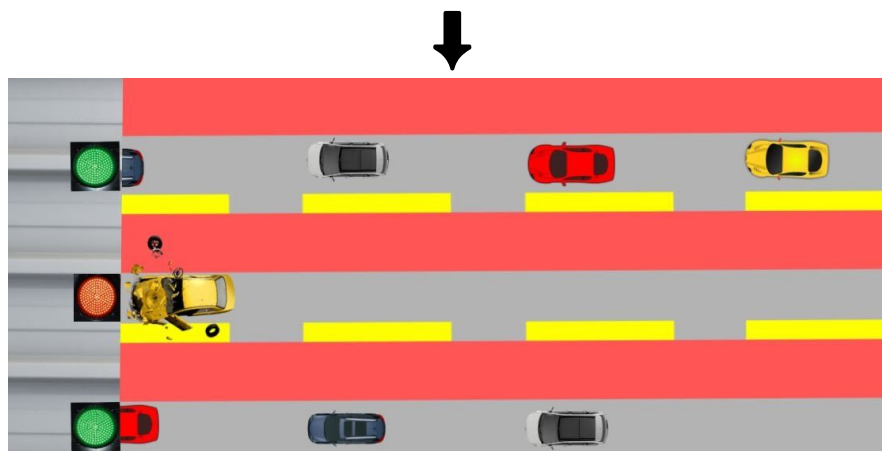
Luego de reconocer cual es la vía que tiene un bloqueo el algoritmo de planificación procederá de diferentes formas.

1. El planificador no asignará más vehículos al carril bloqueado hasta que se desbloquee.
2. Tomando en cuenta que la mayoría de los bloqueos suceden en autos de prioridad normal. El planificador no asignará más vehículos al carril del mismo tipo de prioridad de la vía adyacente que no esté bloqueado y tenga menos vehículos en el momento en el que quedó sucedió el bloqueo.
3. El planificador reparte el vehículo anterior al bloqueo en cada instante a los carriles de igual prioridad de las vías contiguas. Para poder asignarlo a otro carril se evalúa que tenga un espacio disponible para hacerlo y que ningún vehículo le interrumpa el paso.

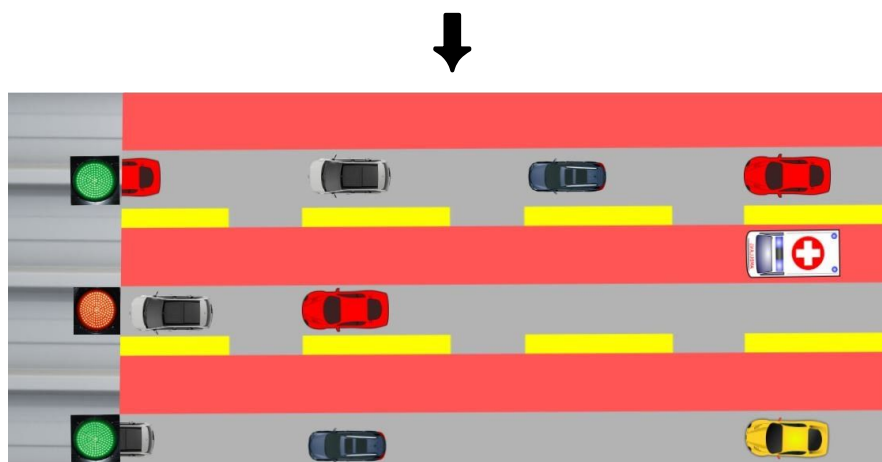
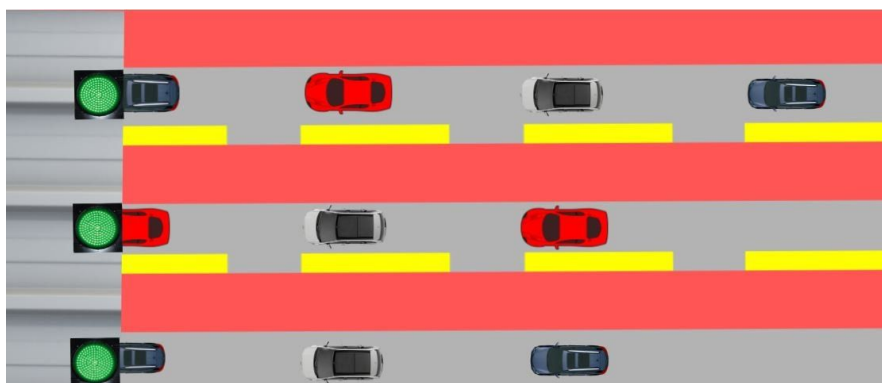
## Diagramas de la solución

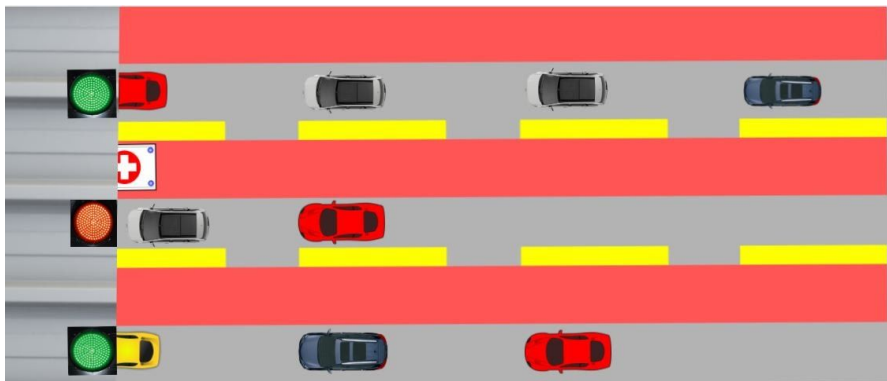
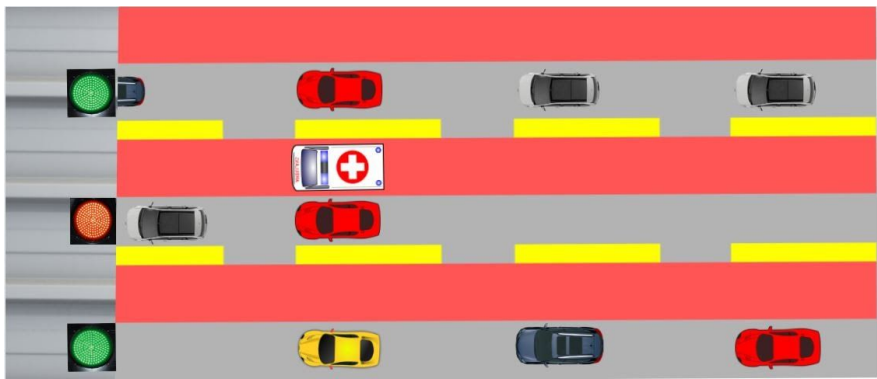
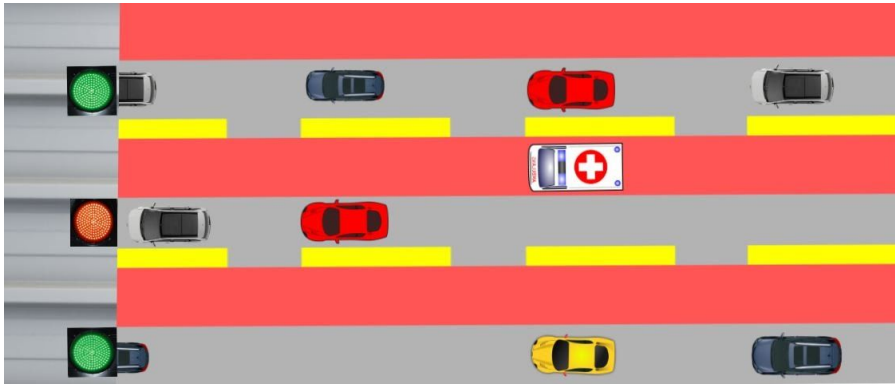
Teniendo en la forma en que se planifica el tráfico, presentamos los siguientes esquemas de un posible caso. En el mismo representamos un escenario en el que un vehículo tiene un accidente y el resto de vehículos tienen que redirigirse a otro carril siempre y cuando ningún otro vehículo le interrumpa el paso.

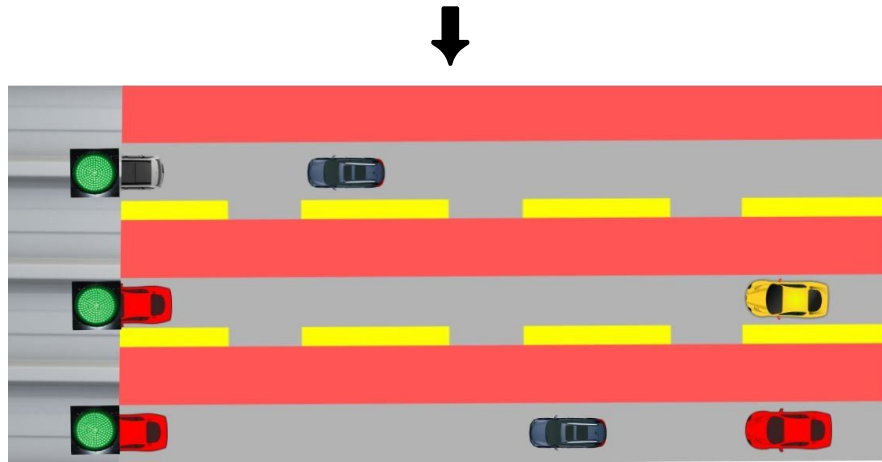
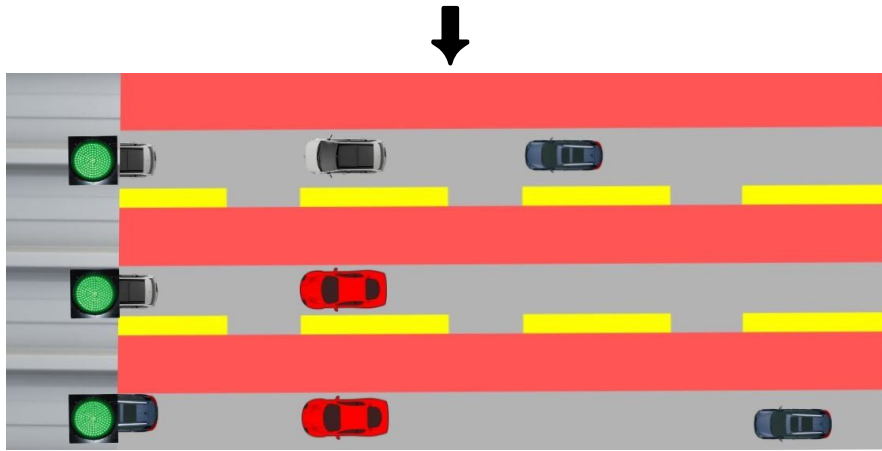




Otra situación clave es cuando llega un vehículo de emergencia al peaje. La secuencia es la que se visualiza a continuación.

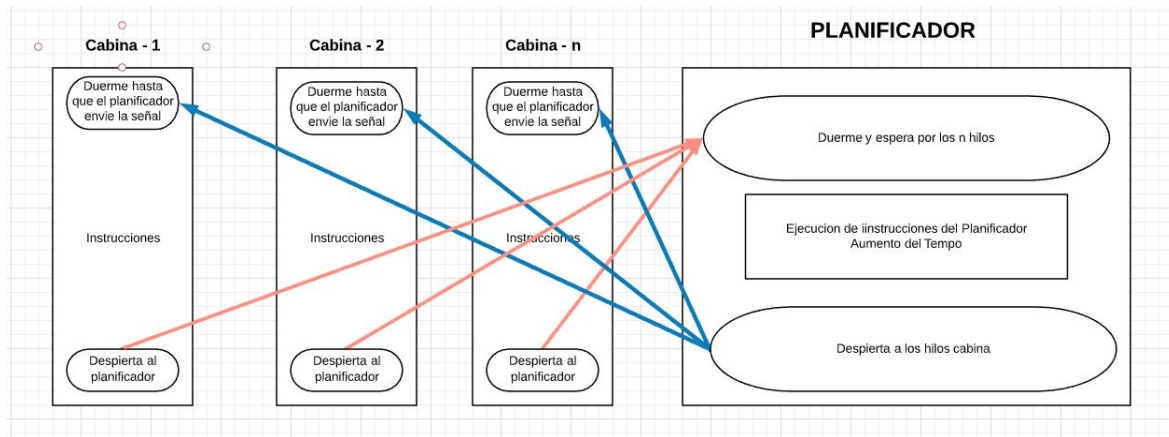




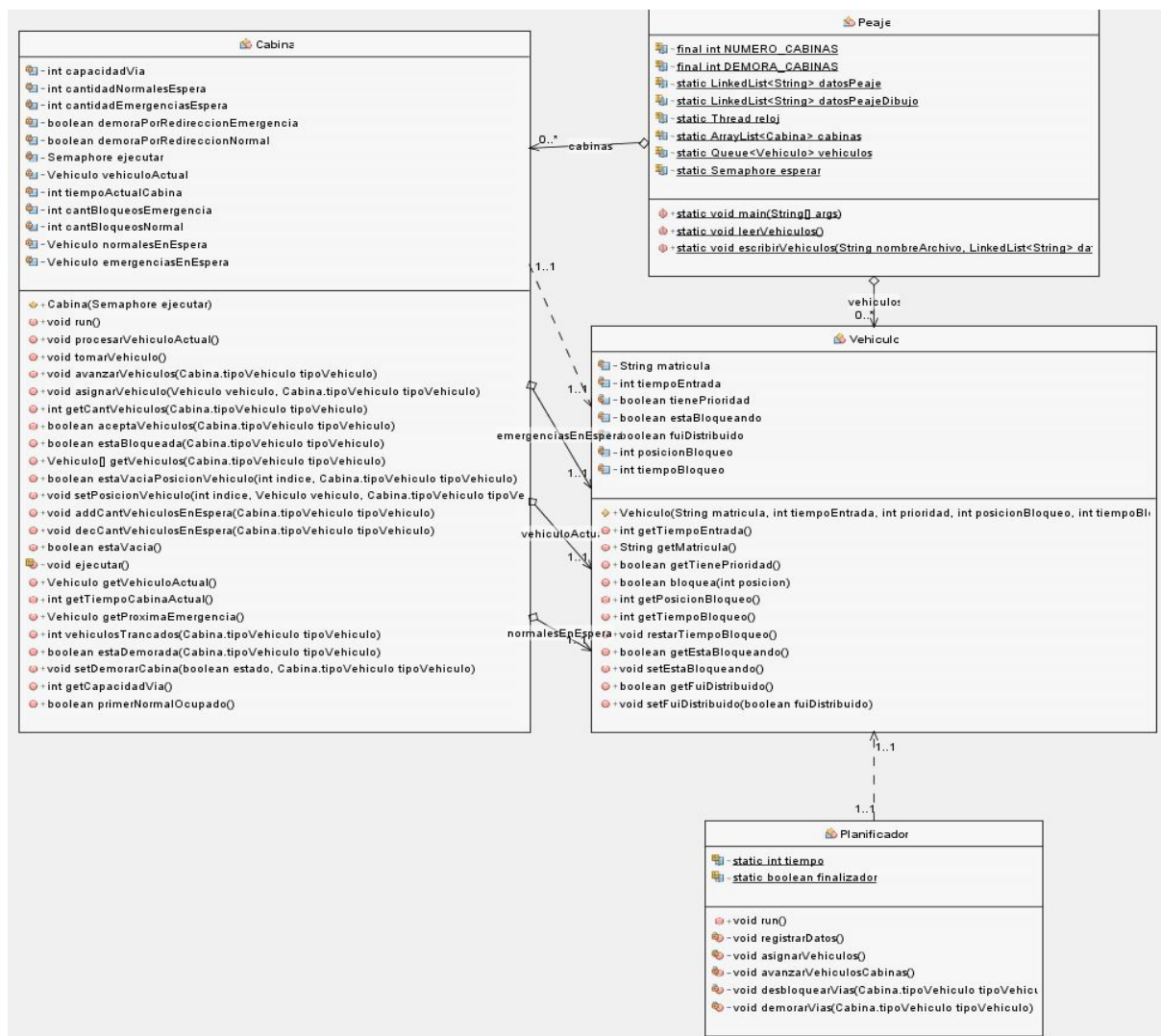


# Diagramas de la implementación

## Diagrama de sincronización



## Diagrama UML





# Simulaciones

## Archivo de entrada

Las simulaciones toman como entrada un archivo separado por comas que representa una determinada cantidad de vehículos que viajan rumbo al peaje.

El archivo contiene en cada línea información asociada al vehículo así como el tiempo en el que llega al peaje y si es un vehículo con prioridad o uno normal , además existen parámetros adicionales para lograr simular los bloqueos.

De esta forma variando los datos de entrada, podemos representar distintos escenarios que pueden ocurrir.

MATRICULA	TIEMPO DE LLEGADA	PRIORIDAD	POSICION DE BLOQUE	DEMORA EN SOLUCIONAR EN BLOQUEO
SBS-5632	1	0	6	500

## Archivos de salida

La simulación genera 2 archivos de salida. Uno con el estado de cada cabina en cada instante de tiempo, y otro con una representación de la distribución y el flujo de los vehículos con respecto al tiempo.

### Ejemplo de una salida con detalle de cabinas

```
Tiempo, Cabina1, Cabina2, Cabina3, Cabina4, Cabina5,
1, vacio, vacio, vacio, vacio, vacio,
2, vacio, vacio, vacio, vacio, vacio,
3, vacio, vacio, vacio, vacio, vacio,
4, vacio, vacio, vacio, vacio, vacio,
5, vacio, vacio, vacio, vacio, vacio,
6, vacio, vacio, vacio, vacio, vacio,
7, vacio, vacio, vacio, vacio, vacio,
8, vacio, vacio, vacio, vacio, vacio,
9, vacio, vacio, vacio, vacio, vacio,
10, vacio, vacio, vacio, vacio, vacio,
11, vacio, vacio, vacio, vacio, vacio,
12, EJECUTA SQL-8815 NORMAL, vacio, vacio, vacio, vacio,
13, EJECUTA SQL-8815 NORMAL, EJECUTA SXI-4795 NORMAL, vacio, vacio, vacio,
14, EJECUTA SQL-8815 NORMAL, EJECUTA SXI-4795 NORMAL, EJECUTA SHA-2249 NORMAL, EJECUTA SEX-8272 NORMAL, vacio,
15, EJECUTA SQL-8815 NORMAL, EJECUTA SXI-4795 NORMAL, EJECUTA SHA-2249 NORMAL, EJECUTA SEX-8272 NORMAL, EJECUTA SBK-5138 NORMAL,
16, EJECUTA SQL-8815 NORMAL, EJECUTA SXI-4795 NORMAL, EJECUTA SHA-2249 NORMAL, EJECUTA SEX-8272 NORMAL, EJECUTA SBK-5138 NORMAL,
17, EJECUTA SAX-5697 NORMAL, EJECUTA SXI-4795 NORMAL, EJECUTA SHA-2249 NORMAL, EJECUTA SEX-8272 NORMAL, EJECUTA SBK-5138 NORMAL,
18, EJECUTA SAX-5697 NORMAL, EJECUTA SEF-6652 NORMAL, EJECUTA SHA-2249 NORMAL, EJECUTA SEX-8272 NORMAL, EJECUTA SBK-5138 NORMAL,
19, EJECUTA SAX-5697 NORMAL, EJECUTA SEF-6652 NORMAL, EJECUTA SVG-5621 NORMAL, EJECUTA SWV-2191 NORMAL, EJECUTA SBK-5138 NORMAL,
20, EJECUTA SAX-5697 NORMAL, EJECUTA SEF-6652 NORMAL, EJECUTA SVG-5621 NORMAL, EJECUTA SWV-2191 NORMAL, EJECUTA SLB-7072 NORMAL,
21, EJECUTA SAX-5697 NORMAL, EJECUTA SEF-6652 NORMAL, EJECUTA SVG-5621 NORMAL, EJECUTA SWV-2191 NORMAL, EJECUTA SLB-7072 NORMAL,
22, EJECUTA SOA-7653 NORMAL, EJECUTA SEF-6652 NORMAL, EJECUTA SVG-5621 NORMAL, EJECUTA SWV-2191 NORMAL, EJECUTA SLB-7072 NORMAL,
23, EJECUTA SOA-7653 NORMAL, EJECUTA SJI-3320 NORMAL, EJECUTA SVG-5621 NORMAL, EJECUTA SWV-2191 NORMAL, EJECUTA SLB-7072 NORMAL,
24, EJECUTA SOA-7653 NORMAL, EJECUTA SJI-3320 NORMAL, EJECUTA SCF-6859 NORMAL, EJECUTA SKY-0160 NORMAL, EJECUTA SLB-7072 NORMAL,
25, EJECUTA SOA-7653 NORMAL, EJECUTA SJI-3320 NORMAL, EJECUTA SCF-6859 NORMAL, EJECUTA SKY-0160 NORMAL, EJECUTA SSS-7811 NORMAL,
26, EJECUTA SOA-7653 NORMAL, EJECUTA SJI-3320 NORMAL, EJECUTA SCF-6859 NORMAL, EJECUTA SKY-0160 NORMAL, EJECUTA SSS-7811 NORMAL,
27, EJECUTA STJ-2229 NORMAL, EJECUTA SJI-3320 NORMAL, EJECUTA SCF-6859 NORMAL, EJECUTA SKY-0160 NORMAL, EJECUTA SSS-7811 NORMAL,
```

## Salida que representa el estado de las colas y las cabinas del peaje

\*\*\*\*\* TIEMPO: 156 \*\*\*\*\*

EMERGENCIAS:	-> CASILLA 1: vacio	vacio , vacio , vacio , vacio , vacio , vacio , SPL-7116 , vacio , vacio , vacio
NORMALES:		SJU-3542 , SCS-7033 , SLT-3650 , SMC-6812 , SMH-5534 , SOL-3562 , SRQ-1053 , SLJ-0451 , SYA-8271 , SVM-7761
EMERGENCIAS:	-> CASILLA 2: SQS-8867	vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio
NORMALES:		SRX-0083 , SYZ-3905 , SFT-7453 , SLL-2713 , SJS-7354 , SOF-4189 , SQX-4485 , SOA-9519 , SZU-1873 , SIE-2035
EMERGENCIAS:	-> CASILLA 3: SHD-4536	vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio
NORMALES:		SEA-1569 , SWO-6828 , SUQ-6270 , SMX-3976 , SFH-5728 , SGJ-7110 , STT-7552 , SPI-8311 , SIG-0899 , SWD-6638
EMERGENCIAS:	-> CASILLA 4: SLL-1916	vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio
NORMALES:		SZB-8546 , SMA-8975 , SCD-7680 , SSF-6282 , SJQ-2447 , SUN-4062 , SJV-0611 , SXQ-4271 , SLC-2055 , SIC-5334
EMERGENCIAS:	-> CASILLA 5: SDX-9854	vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio
NORMALES:		SCT-1098 , SDA-3110 , SQJ-1611 , SIU-6965 , SPJ-6658 , SDW-9978 , SPZ-2451 , SVA-1495 , SOT-8591 , SFC-4481

# Comprobación del comportamiento

## Distribución del tráfico

- Para casillas que no están bloqueadas siempre se asignan los vehículos de cierta prioridad a la casilla menos sobrecargada por automóviles con dichas características:

Ejemplo para vehículos con prioridad y 2 cabinas:

- Entrada:

SBS-5632,1,0,0,0  
AHN-6303,2,0,0,0  
TYC-9071,3,0,0,0  
FBE-5382,4,0,0,0

- Salida:

***** TIEMPO: 1 *****											
EMERGENCIAS:			-> CASILLA 1: vacio								
NORMALES:			vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , SBS-5632								
EMERGENCIAS:			-> CASILLA 2: vacio								
NORMALES:			vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio								
***** TIEMPO: 2 *****											
EMERGENCIAS:			-> CASILLA 1: vacio								
NORMALES:			vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , SBS-5632 , vacio								
EMERGENCIAS:			-> CASILLA 2: vacio								
NORMALES:			vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , AHN-6303								
***** TIEMPO: 3 *****											
EMERGENCIAS:			-> CASILLA 1: vacio								
NORMALES:			vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , SBS-5632 , vacio , TYC-9071								
EMERGENCIAS:			-> CASILLA 2: vacio								
NORMALES:			vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , AHN-6303 , vacio								
***** TIEMPO: 4 *****											
EMERGENCIAS:			-> CASILLA 1: vacio								
NORMALES:			vacio , vacio , vacio , vacio , vacio , vacio , SBS-5632 , vacio , TYC-9071 , vacio								
EMERGENCIAS:			-> CASILLA 2: vacio								
NORMALES:			vacio , vacio , vacio , vacio , vacio , vacio , vacio , AHN-6303 , vacio , FBE-5382								

El comportamiento es igual para vehículos sin prioridad y otro número de cabinas.

- Para vehículos sin importar su prioridad, no se hacen asignaciones de vehículos a casillas con su correspondiente carril bloqueado:

Ejemplo para vehículos sin prioridad y 2 cabinas:

- Entrada:
 

SBS-5632,1,0,8,50  
 AHN-6303,1,0,0,0  
 TYC-9071,1,0,0,0  
 ZBS-5632,1,0,0,0
- Salida:

***** TIEMPO: 1 *****										
EMERGENCIAS:	-> CASILLA 1:	vacio		vacio	,	vacio	,	vacio	,	vacio
NORMALES:				vacio	,	vacio	,	vacio	,	vacio
EMERGENCIAS:	-> CASILLA 2:	vacio		vacio	,	vacio	,	vacio	,	vacio
NORMALES:				vacio	,	vacio	,	vacio	,	SB
***** TIEMPO: 2 *****										
EMERGENCIAS:	-> CASILLA 1:	vacio		vacio	,	vacio	,	vacio	,	vacio
NORMALES:				vacio	,	vacio	,	vacio	,	SB
EMERGENCIAS:	-> CASILLA 2:	vacio		vacio	,	vacio	,	vacio	,	vacio
NORMALES:				vacio	,	vacio	,	vacio	,	SB
***** TIEMPO: 3 *****										
EMERGENCIAS:	-> CASILLA 1:	vacio		vacio	,	vacio	,	vacio	,	vacio
NORMALES:				vacio	,	vacio	,	vacio	,	SB
EMERGENCIAS:	-> CASILLA 2:	vacio		vacio	,	vacio	,	vacio	,	vacio
NORMALES:				vacio	,	vacio	,	vacio	,	SB
***** TIEMPO: 4 *****										
EMERGENCIAS:	-> CASILLA 1:	vacio		vacio	,	vacio	,	vacio	,	vacio
NORMALES:				vacio	,	vacio	,	vacio	,	SB
EMERGENCIAS:	-> CASILLA 2:	vacio		vacio	,	vacio	,	vacio	,	vacio
NORMALES:				vacio	,	vacio	,	vacio	,	SB

El comportamiento es igual para vehículos con prioridad y otro número de cabinas.



- No se asignan vehículos a un carril de una cierto nivel de prioridad de una vía si, en el instante del corte, sin estar trancada es la que tiene menos automóviles de ese tipo de las adyacentes a una vía con su correspondiente carril bloqueado. En dicho caso se volverán a asignar vehículos luego de que se le dirijan todos los automóviles con esa prioridad trancados por el bloqueo del otro carril.

Ejemplo para vehículos sin prioridad y 3 cabinas:

- Entrada:

SBS-5632,1,0,0,0  
 AHN-6303,1,0,5,50  
 TYC-9071,2,0,8,2  
 FBE-5382,2,0,0,0  
 ACF-6859,3,0,0,0  
 SAT-5208,3,0,0,0  
 SCV-1677,4,0,0,0  
 EKY-0160,7,0,0,0

- Salida:

***** TIEMPO: 6 *****																						
EMERGENCIAS:	-> CASILLA 1:	vacio		vacio	,	vacio	,	vacio	,	vacio	,	vacio										
NORMALES:				vacio	,	vacio	,	vacio	,	SBS-5632	,	FBE-5382	,	SAT-5208	,	vacio	,	vacio	,	vacio		
EMERGENCIAS:	-> CASILLA 2:	vacio		vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio		
NORMALES:				vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	AHN-6303 B	,	ACF-6859	,	SCV-1677	,	vacio	,	vacio
EMERGENCIAS:	-> CASILLA 3:	vacio		vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio		
NORMALES:				vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	TYC-9071	,	vacio	,	vacio
***** TIEMPO: 7 *****																						
EMERGENCIAS:	-> CASILLA 1:	vacio		vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio		
NORMALES:				vacio	,	vacio	,	vacio	,	SBS-5632	,	FBE-5382	,	SAT-5208	,	vacio	,	vacio	,	vacio	,	EKY-0160
EMERGENCIAS:	-> CASILLA 2:	vacio		vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio		
NORMALES:				vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	AHN-6303 B	,	SCV-1677	,	vacio	,	vacio	,	vacio
EMERGENCIAS:	-> CASILLA 3:	vacio		vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	vacio		
NORMALES:				vacio	,	vacio	,	vacio	,	vacio	,	vacio	,	ACF-6859	,	TYC-9071	,	vacio	,	vacio	,	vacio

Se puede observar que en el tiempo 7, en condiciones normales, el vehículo “EKY-0160” debería ser asignado a la cabina 3 por ser la que tiene menos vehículos de prioridad normal. Sin embargo, cómo no se pueden asignar vehículos al carril de la cabina adyacente con menos vehículos normales en el instante de bloqueo hasta que se repartan los automóviles trancados, “EKY-0160” se envía a la cabina 1.

El comportamiento es análogo para vehículos con prioridad y distinto número de cabinas. Además si se trata de una cabina lateral, se deja de asignar a la única cabina que tiene a su lado.

El comportamiento es igual para vehículos con prioridad y otro número de cabinas.

### *Priorización del paso para los vehículos de emergencia*

- En el momento que un vehículo de emergencia es asignado a una vía el semáforo de aviso de emergencia se enciende indicando a los conductores que esperan para acceder a dicha cabina que deben esperar a que se atienda primero al vehículo con prioridad de paso. Luego de que el vehículo es atendido, si no hay más automóviles de prioridad en espera, el semáforo vuelve a cambiar y el tránsito por esta vía vuelve a la normalidad.

Ejemplo con 1 cabina:

- Entrada:

SBS-5632,1,0,0,0  
AHN-6303,1,0,0,0  
TYC-9071,2,0,0,0  
FBE-5382,2,0,0,0  
ACF-6859,3,0,0,0  
SAT-5208,3,0,0,0  
SCV-1677,4,1,0,0

- Salida:

```
***** TIEMPO: 5 *****
EMERGENCIAS: -> CASILLA 1: vacio | vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , SCV-1677
NORMALES:      | vacio , vacio , vacio , vacio , vacio , SBS-5632 , AHN-6303 , TYC-9071 , FBE-5382 , ACF-6859
***** TIEMPO: 9 *****
EMERGENCIAS: -> CASILLA 1: vacio | vacio , vacio , vacio , vacio , vacio , SCV-1677 , vacio , vacio , vacio , vacio , vacio
NORMALES:      | vacio , SBS-5632 , AHN-6303 , TYC-9071 , FBE-5382 , ACF-6859 , vacio , vacio , vacio , vacio , vacio
***** TIEMPO: 11 *****
EMERGENCIAS: -> CASILLA 1: vacio | vacio , vacio , vacio , SCV-1677 , vacio , vacio , vacio , vacio , vacio , vacio , vacio
NORMALES:      | SBS-5632 , AHN-6303 , TYC-9071 , FBE-5382 , ACF-6859 , vacio , vacio , vacio , vacio , vacio
***** TIEMPO: 14 *****
EMERGENCIAS: -> CASILLA 1: vacio | SCV-1677 , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio
NORMALES:      | SBS-5632 , AHN-6303 , TYC-9071 , FBE-5382 , ACF-6859 , vacio , vacio , vacio , vacio , vacio
***** TIEMPO: 15 *****
EMERGENCIAS: -> CASILLA 1: SCV-1677 | vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio
NORMALES:      | SBS-5632 , AHN-6303 , TYC-9071 , FBE-5382 , ACF-6859 , vacio , vacio , vacio , vacio , vacio
***** TIEMPO: 18 *****
EMERGENCIAS: -> CASILLA 1: SBS-5632 | vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio , vacio
NORMALES:      | AHN-6303 , TYC-9071 , FBE-5382 , ACF-6859 , vacio , vacio , vacio , vacio , vacio , vacio
```

Se puede observar cómo a pesar de que hay vehículos normales que llegan antes a las inmediaciones de la cabina, se deja pasar antes a la emergencia que llega después. El comportamiento es análogo para distinto número de cabinas.

## Bloqueo en una vía

- Se permite que la totalidad de los automóviles que estaban en la vía que ocurrió el bloqueo puedan redistribuirse a las vías contiguas. Se toma en cuenta que un vehículo debe esperar para cambiarse de vía a tener un espacio para hacerlo en el nuevo carril y que también haya espacio en el carril que debe atravesar para llegar hasta allí de forma tal que no se genere un accidente.

Ejemplo para vehículos sin prioridad:

- Entrada:

SBS-5632,1,1,0,0  
 AHN-6303,1,0,7,10  
 TYC-9071,2,0,0,0  
 FBE-5382,2,0,0,0  
 ACF-6859,3,0,0,0  
 SCV-1677,4,1,0,0

- Salida:

***** TIEMPO: 4 *****										
EMERGENCIAS:	-> CASILLA 1:	vacio	vacio	vacio	vacio	vacio	vacio	SBS-5632	vacio	vacio
NORMALES:			vacio	vacio	vacio	vacio	vacio	SLJ-0451	TYC-9071	ACF-6859
EMERGENCIAS:	-> CASILLA 2:	vacio	vacio	vacio	vacio	vacio	vacio	vacio	vacio	SCV-1677
NORMALES:			vacio	vacio	vacio	vacio	vacio	vacío	AHN-6303 B	FBE-5382
***** TIEMPO: 5 *****										
EMERGENCIAS:	-> CASILLA 1:	vacio	vacio	vacio	vacio	vacio	vacio	SBS-5632	vacio	vacio
NORMALES:			vacio	vacio	vacio	vacio	vacio	SLJ-0451	TYC-9071	ACF-6859
EMERGENCIAS:	-> CASILLA 2:	vacio	vacio	vacio	vacio	vacio	vacio	vacio	SCV-1677	vacio
NORMALES:			vacio	vacio	vacio	vacio	vacio	vacío	AHN-6303 B	FBE-5382
***** TIEMPO: 6 *****										
EMERGENCIAS:	-> CASILLA 1:	vacio	vacio	vacio	vacio	vacio	vacio	SBS-5632	vacio	vacio
NORMALES:			vacio	vacio	vacio	vacio	vacio	SLJ-0451	TYC-9071	ACF-6859
EMERGENCIAS:	-> CASILLA 2:	vacio	vacio	vacio	vacio	vacio	vacio	vacío	SCV-1677	vacio
NORMALES:			vacio	vacio	vacio	vacio	vacio	vacío	AHN-6303 B	FBE-5382
***** TIEMPO: 7 *****										
EMERGENCIAS:	-> CASILLA 1:	vacio	vacio	vacio	SBS-5632	vacio	vacio	vacio	vacio	vacio
NORMALES:			vacio	vacio	SLJ-0451	TYC-9071	ACF-6859	vacio	FBE-5382	vacio
EMERGENCIAS:	-> CASILLA 2:	vacio	vacio	vacio	vacio	vacio	vacio	SCV-1677	vacio	vacio
NORMALES:			vacio	vacio	vacio	vacio	vacio	vacío	AHN-6303 B	vacio

Se puede observar como el vehículo “FBE-5382” no puede desviarse a la cabina 1 en el tiempo 4 porque a pesar de tener espacio libre a su lado en el carril de emergencia de la vía de la otra cabina, el espacio al que debería moverse en el carril normal está ocupado por el automóvil “TYC-9071”. En el tiempo 5 si tiene el espacio mencionado pero esta vez el vehículo “SCV-1677” no lo deja pasar hasta allí dado que tiene que cruzar dicho carril. Recién en el tiempo 6 se dan ambas condiciones y se hace el movimiento que se refleja en el tiempo 7.

El comportamiento es igual para vehículos con prioridad y otro número de cabinas.

## Casos simulados

### 1. Alta densidad de tráfico

#### 1.1.1 Escenario ideal

En este escenario se simula el pasaje de una gran cantidad de vehículos que llegan en un corto lapso de tiempo pero sin ningún evento de bloqueo o emergencia, el algoritmo de planificación ordena los vehículos distribuyendolos en las colas con menor carga resultando en un tiempo total de procesamiento de 214 unidades de tiempo.

Parámetros:

Cantidad de Vehículos	199
Cabinas	5
Procesamiento en cabina	5
Archivo de entrada	CargaAlta.csv
Archivos de salida	SalidaCargaAlta_5c.csv SalidaCargaAlta_5c.txt
Resultado	214

#### 1.1.2 Escenario ideal con aumento de cabinas

En base a datos históricos se puede planificar la maximización de cabinas habilitadas para atender el tráfico esperado, se hace una simulación con 10 cabinas resultando en un tiempo de procesamiento de 135 unidades de tiempo.

Parámetros:

Cantidad de Vehículos	199
Cabinas	10
Procesamiento en cabina	5
Archivo de entrada	EntradaCargaAlta.csv
Archivos de salida	SalidaCargaAlta_10c.csv SalidaCargaAlta_10c.txt
Resultado	135



### 1.2 Escenario con Emergencias

A la misma carga de tráfico y con 5 cabinas se le añaden 8 vehículos con alta prioridad. Al reconocer los vehículos de alta prioridad el algoritmo de planificación primero los asigna a una vía y avisa a la misma para que esta detenga la atención a los otros vehículos hasta que la emergencia sea atendida.

Se puede observar que el tiempo de proceso es de 231 unidades, por lo que existe una leve penalización con respecto al caso sin emergencias que arroja un resultado de 214 unidades de tiempo.

Cantidad de Vehículos	199
Cabinas	5
Procesamiento en cabina	5
Vehículos con prioridad	8
Archivo de entrada	EntradaCargaAlta.csv
Archivos de salida	SalidaCargaAlta_5c_8p.csv SalidaCargaAlta_5c_8p.txt
Resultado	231

### 1.3 Escenario con Bloqueos y Emergencias

A la misma carga con las 5 cabinas y los vehículos de prioridad se agregan eventos que generan bloqueos en 3 puntos en distintos momentos con una demora de 30 unidades de tiempo cada uno para dar solución al bloqueo el algoritmo deja de enviar vehículos al carril afectado e identifica el carril adyacente con menor carga al cual suspende la asignación de vehículos con el fin de desagotar los vehículos estancados por el bloque. Se puede apreciar que la penalización es pequeña a pesar de tener fuera de servicio los carriles por un tiempo considerable

Cantidad de Vehículos	199
Cabinas	5
Procesamiento en cabina	5
Bloqueos	3
Tipo de bloqueo	30
Archivo de entrada	EntradaCargaAlta.csv
Archivos de salida	SalidaCargaAlta_5c_8p_3b.csv SalidaCargaAlta_5c_8p_3b.txt
Resultado	236

## 2. Baja densidad de tráfico

### 2.1.1 Escenario ideal

En este escenario los vehículos llegan con poca frecuencia, sin ninguna clase de evento de bloqueo o emergencia. Dado que esta situación se traduce en pocos vehículos llegando, el planificador no tendrá más que ejecutar su comportamiento habitual, distribuyendo el tráfico entre las cabinas con menor carga. Para este caso particular se utilizaron un total de 20 vehículos, resultando en un tiempo de proceso de 92.

Cantidad de vehículos	20
Cabinas	2
Demora por cabina	2
Archivo de entrada	CargaBaja.csv
Archivos de salida	CargaBaja_2c.csv CargaBaja_2c.txt
Instante de llegada del primer vehículo	1
Instante de llegada del último vehículo	80
Resultado	92

### 2.1.2 Escenario ideal con aumento de cabinas

En este caso se realiza un aumento en la cantidad de cabinas utilizadas, de 2 a 4.

Aquí el algoritmo de planificación tiene más lugares a donde asignar los vehículos que llegan, pero sin embargo el tiempo de procesamiento permanece igual que en el caso en que se utilizaron 2 cabinas.

Se puede ver entonces que ante poco tráfico, no es necesario hacer un aumento de cabinas.

Cantidad de vehículos	20
Cabinas	4
Demora por cabina	2
Archivo de entrada	CargaBaja.csv
Archivos de salida	CargaBaja_4c.csv CargaBaja_4c.txt
Instante de llegada del primer vehículo	1
Instante de llegada del último vehículo	80
Resultado	92

## 2.2 Escenario con emergencias

Con la misma carga de tráfico y cabinas se añaden 3 vehículos con alta prioridad. En el momento en que el algoritmo de planificación detecta los vehículos en emergencia le asigna una vía y detiene el flujo de vehículos para que pueda pasar sin problemas. Se puede observar que el tiempo de proceso continúa siendo de 92 unidades, lo que es coherente por la baja carga de vehículos y su distribución en el tiempo.

Cantidad de vehículos	20
Cabinas	2
Demora por cabina	2
Vehículos con prioridad	3
Archivo de entrada	CargaBaja.csv
Archivos de salida	CargaBaja_2c_3p.csv CargaBaja_2c_3p.txt
Instante de llegada del primer vehículo	1
Instante de llegada del último vehículo	80
Resultado	92

## 2.3 Escenario con Bloqueos y Emergencias

Con la misma carga de tráfico y cabinas se añaden 3 vehículos que presentan desperfectos en algún tramo, con demoras de 10, 5 y 7 unidades de tiempo respectivamente. En este escenario particular el algoritmo de planificación debe resolver el problema de distribuir los vehículos de forma equitativa teniendo en cuenta que en los carriles pueden estar ocurriendo distintos eventos. En los casos en los que hay un bloqueo de vía, se detiene la asignación de vehículos al carril afectado y busca el carril adyacente que tenga la menor carga. A su vez también detiene la asignación de vehículos a la vía destino con el fin de desagotar los que estén y permitir cambiar de vía a los que están trancados.

Para este caso en particular se observa una pequeña penalización de una unidad de tiempo, lo que podría verse explicado por el tiempo que le lleva a los vehículos accidentados volver a funcionar.

Cantidad de vehículos	20
Cabinas	2
Demora por cabina	2
Vehículos con prioridad	3

Vehículos bloqueados	3
Archivo de entrada	CargaBaja.csv
Archivos de salida	CargaBaja_2c_3p_3b.csv CargaBaja_2c_3p_3b.txt
Instante de llegada del primer vehículo	1
Instante de llegada del último vehículo	80
Resultado	93

# Estructura de la implementación

## Clase: Cabina

Es la clase con la que se representa una cabina de cobro.

### Atributos

- *capacidadVia*  
Es la longitud máxima de la cola que tiene una vía.
- *cantidadNormalesEspera*  
Es la cantidad de vehículos sin prioridad que están circulando por esta vía.
- *cantidadEmergenciasEspera*  
Es la cantidad de vehículos con prioridad que están circulando por esta vía.
- *demoraPorRedireccionEmergencia*  
Bandera que indica si la vía está siendo demorada para admitir redirección de vehículos con prioridad desde otras vías.
- *demoraPorRedireccionNormal*  
Bandera que indica si la vía está siendo demorada para admitir redirección de vehículos normales desde otras vías.
- *ejecutar*  
Semáforo utilizado para la sincronización con el planificador.
- *vehiculoActual*  
El vehículo que está procesando actualmente la cabina.
- *tiempoActualCabina*  
El tiempo que lleva procesando el vehículo actual.
- *cantBloqueosEmergencia*  
Cantidad de posiciones que tienen un bloqueo en la vía de emergencias.
- *cantBloqueosNormal*  
Cantidad de posiciones que tienen un bloqueo en la vía normal.
- *normalesEnEspera*  
Cola de vehículos normales.
- *emergenciasEnEspera*  
Cola de vehículos con prioridad.

## Métodos

- *run*
- *procesarVehiculoActual*  
Realiza las tareas correspondientes al proceso del vehículo actual si el tiempo de proceso a concluido libera la cabina.
- *tomarVehiculo*  
Si la cabina está vacía toma el vehículo siguiente a procesar.
- *avanzarVehiculos*  
Realiza el desplazamiento de los vehiculos que estan en las colas,teniendo en cuenta si hay bloqueos que no permitan el avance.
- *asignarVehiculo*  
Asignación de un nuevo vehículo a una de las colas de esta vía.
- *getCantVehiculos*  
Devuelve la cantidad de vehículos que hay actualmente en esta vía.
- *aceptaVehiculos*  
Recibe el tipo de vehículo como parámetro y retorna verdadero si la cola de este tipo acepta nuevos vehículos.
- *estaBloqueada*  
Recibe el tipo de vehículo como parámetro y retorna verdadero si la cola de este tipo tiene algun bloqueo.
- *getVehiculos*  
Recibe el tipo de vehículo y retorna la lista de vehículos de este tipo.
- *estaVaciaPosicionVehiculo*  
Recibe una posición y un tipo de vehículo y retorna verdadero si está vacía la cola en esa posicion.
- *setPosicionVehiculo*  
Recibe una posición, un tipo de vehículo y un vehículo, luego asigna el vehículo a la posición recibida en la cola correspondiente.
- *addCantVehiculosEnEspera*  
Aumente el contador de vehículos en espera
- *decCantVehiculosEnEspera*  
Decrementa el contador de vehículos en espera
- *estaVacia*  
Recibe el tipo de vehículo y retorna verdadero si la cola de este tipo está vacía.
- *ejecutar*  
Libera el semáforo que permite la ejecución de este hilo.
- *getVehiculoActual*  
Retorna el vehículo que tiene actualmente la cabina.
- *getTiempoCabinaActual*  
Retorna el tiempo de procesamiento transcurrido para el vehículo que ocupa la cabina actualmente.

- *getProximaEmergencia*
- *vehiculosTrancados*  
Recibe el tipo de vehículo y retorna la cantidad de vehiculos que estan estancados producto de un bloqueo en la cola correspondiente.
- *estaDemorada*  
Retorna verdadero si la cabina está siendo utilizada para desviar tráfico desde otras vías.
- *setDemorarCabina*  
Recibe el vehículo y el valor de la bandera de demora, luego setea el valor de la bandera correspondiente a la cola del tipo recibido.
- *getCapacidadVia*  
Retorna la capacidad de la vía.
- *primerNormalOcupado*

## Clase: Peaje

### Atributos

- NUMERO\_CABINAS  
Contiene el número de cabinas operando en el peaje.
- DEMORA\_CABINAS  
Contiene el tiempo que demora una cabina en procesar un vehículo.
- *datosPeaje*  
Variable de almacenaje de datos para los archivos de salida.
- *datosPeajeDibujo*  
Variable de almacenaje de datos para los archivos de salida.
- *reloj*  
Contador de los instantes de tiempo transcurridos.
- *cabinas*  
Lista con todas las cabinas existentes.
- *vehículos*  
Cola de autos que se ejecutarán.
- *esperar*  
Semáforo utilizado para sincronizar.

## Métodos

- *leerVehiculos*  
Procesa el archivo de entrada.
- *escribirVehiculos*  
Escribe los archivos de salida.

## Clase: Planificador

### Atributos

- *tiempo*  
Contador del tiempo transcurrido.
- *finalizador*  
Bandera que indica que el programa llega a su fin.

### Métodos

- *run*
- *registrarDatos*  
Genera las estructuras para los archivos de salida en base a los datos obtenidos en cada instante de tiempo.
- *asignarVehiculos*  
Es el método que se encarga de distribuir los vehículos que llegan al peaje.
- *avanzarVehiculosCabinas*  
Indica a cada cabina que tiene que avanzar sus vehículos.
- *desbloquearVias*  
Resuelve los bloqueos de de las vías redirigiendo vehículos a las vías adyacentes.
- *demorarVias*  
Resuelve qué vías van a ser demoradas con el fin de distribuir los vehículos trancados.

## Clase: Vehículo

### Atributos

- *matricula*  
Número de matrícula del vehículo.
- *tiempoEntrada*  
Indica el instante de tiempo en el que debería llegar el vehículo al peaje si ningún obstaculo se lo impide.



- *tienePrioridad*  
Indica si el vehículo es de prioridad.
- *estaBloqueando*  
Indica si el vehículo sufrió un desperfecto y no puede circular.
- *fuiDistribuido*  
Indica si el vehículo fue distribuido a una vía adyacente.
- *posicionBloqueo*  
Contiene la posición en la que el vehículo sufre un desperfecto.
- *tiempoBloqueo*  
Contiene el tiempo que el vehículo permanecerá bloqueado.

## Métodos

- *getTiempoEntrada*  
Devuelve el tiempo de entrada del vehículo.
- *getMatricula*  
Devuelve la matrícula del vehículo.
- *getTienePrioridad*  
Devuelve verdadero si el vehículo es de prioridad.
- *bloquea*  
Devuelve verdadero si el vehículo se encuentra bloqueando la posición.
- *getPosicionBloqueo*  
Devuelve la posición en la que el vehículo se bloquea.
- *getTiempoBloqueo*  
Devuelve la el tiempo que el vehículo permanece bloqueado.
- *restarTiempoBloqueo*  
Disminuye en una unidad el tiempo de bloqueo
- *getEstaBloqueando*  
Devuelve verdadero si el vehículo está bloqueando.
- *setEstaBloqueando*  
Setea como verdadera la bandera que indica que el vehículo está bloqueando.
- *getFuiDistribuido*  
Devuelve el valor de la bandera que indica si el vehículo fue distribuido en el último tiempo.
- *setFuiDistribuido*  
Setea el valor de la bandera que indica si el vehículo fue distribuido en el último tiempo.

# Conclusiones

En la extensión de este proyecto tuvimos la oportunidad de poner en práctica conocimientos obtenidos en clase. Los mismos en algún sentido, pueden llegar a ser bastante teóricos y abstractos, por lo que fue una buena oportunidad para clarificar lo que en primera instancia era bastante complejo.

El desarrollo de programas multihilo es muy importante para representar situaciones que ocurren en la realidad en las que se compite o se requiere sincronización para el acceso de recursos entre diferentes entidades. El ejercicio de practicar haciendo paralelismos con problemas de la vida real para nosotros, y encontrar los puntos en común para hacer una solución, fue muy útil para entender mejor muchos aspectos que de otra forma hubieran quedado en la teoría y no los habríamos comprendido de forma correcta.

Diseñando la solución arriba detallada, pudimos darnos cuenta cómo funcionan los planificadores de un determinado sistema, como sincronizar distintos procesos que podrían llegar a tener conflictos y que toda solución que mejora algún aspecto, puede llegar a tener su contrapartida negativa.

Por medio de las métricas obtenidas pudimos comprobar el impacto que tiene la planificación en la optimización de la solución de un sistema. Vimos que en muchos casos, un pequeño cambio en la forma de atacar una problemática particular puede llegar a tener un gran impacto en los resultados finales.

## Lista de referencias

1. Telepeaje (2020). ¿Cómo funciona telepeaje?. Montevideo, UY. Rincón 258, piso 5 (Corporación vial del Uruguay). Recuperado de <https://www.telepeaje.com.uy/>
2. Alvaro Ferreiro (2017). Desde arriba - Peaje Pando - Arroyo Pando - Neptunia. Canelones, UY. Recuperado de <https://youtu.be/q-wc6OSslks>
3. Google (s.f). Mapa de Peaje de Pando, Canelones, Uruguay en Google Maps. Recuperado en Julio de 2015 de: <https://www.google.com/maps/place/Peaje+Pando/@-34.7848582,-55.8878306,13z/data=!4m5!3m4!1s0x0:0x74dcb58947a3cfe!8m2!3d-34.7848359!4d-55.888449?shorturl=1>
4. Tanenbaum, A. (2009). *Sistemas operativos modernos* (3ª ed.). México: Pearson Educación.
5. Silberschatz, A., Galvin, P., Gagne, G. (2006). *Fundamentos de sistemas operativos* (7ª ed.). España: McGraw-Hill.
6. Department of Computer Science, University of Illinois at Chicago (s.f). *Operating Systems: Threads*. Chicago, USA. Recuperado el 16 de Mayo de 2020, de [https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/4\\_Threads.html](https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/4_Threads.html)